SMALL SATELLITE
PORTFOLIO
AIR FORCE RESEARCH LABORATORY

We Consistently Miscommunicate Risk

Dr. Lee Jasper

What is Risk?

- **Risk = anything that may negatively impact mission's technical or programmatic capability**
  - Anything that might cause the program to adjust schedule, cost, staffing, or scope

- Paranoia = tangential or low likelihood events; not something to plan around

Why Do We Capture Risk?

1. Understand possible issues/problems that could occur

2. Communicate to team & stakeholders items of concern

3. Logically decide what to do about issues/problems
   - Mitigation is a trade space
   - The team should use the opportunity to engage stakeholders

CHALLENGES

Often downplayed
Often not fully understood

Hard to decide without proper communication
Can be hard to prioritize

Trust is fundamental to be earned both from stakeholders and team

Brief Case Study (1/3)

- Scenario: In a recent mission the risk of not receiving frequency licensing by launch looked high
- Risk: Conveyed as the key mission risk for entirety of program life
- Trust: Partners conveyed this risk getting realized up *their* management chain
  - Management, not having trust in the local team assumed this was probable mission ending, tried to ground the mission
  - Significant high level interaction, interpersonal information sharing, trust building was required before management convinced to take/accept risk as operations schedule limitation + low likelihood chance of mission loss

Risk never changed nor did mitigation plan, only trust building

Brief Case Study (2/3)

- Scenario: In a recent mission the risk of not receiving frequency licensing by launch looked high
- Risk: Conveyed as the key mission risk for entirety of program life
- Trust: Partners conveyed this risk getting realized up *their* management chain
  - Management, not having trust in the local team assumed this was probable mission ending, tried to ground the mission
  - Significant high level interaction, interpersonal information sharing, trust building was required before management convinced to take/accept risk as operations schedule limitation + low likelihood chance of mission loss

If so concerning, why wasn't this better addressed?

Risk never changed nor did mitigation plan, only trust building

Brief Case Study (3/3)

- Scenario: In a recent mission the risk of not receiving frequency licensing by launch looked high
- Risk: Conveyed as the key mission risk for entirety of program life
- Trust: Partners conveyed this risk getting realized up *their* management chain
  - Management, not having trust in the local team assumed this was probable mission ending, tried to ground the mission
  - Significant high level interaction, interpersonal information sharing, trust building was required before management convinced to take/accept risk as operations schedule limitation + low likelihood chance of mission loss

Various levels of communication and trust are necessary

Risk never changed nor did mitigation plan, only trust building

Miscommunicating Risk

- Risks can be challenging to understand if:
  - You're not directly working the risk (technical)
  - You don't understand the larger environment (Policy, priority, launch, etc.)
- Much of this has been wrapped up in documenting…
  - *Good documentation is important to understand risk and risk mitigation, but the main point of mission assurance is to improve the chances of mission success, not to document risks* (-Barbara Braun, Agile Mission Assurance)
- What we should be doing:
  - (1) Understand, (2) Communicate, (3) Logically decide action
  - Team:
    - Be honest about concerns
    - Guide discussion for mitigation trade-space; may include guidance on scope changes or what NOT to do ("go up the chain" doesn't help much)
  - Stakeholders:
    - Push decision authority down / enable team (ask "When do you need my help on this?")
    - Understand and convey their limitations for mitigations (budget, policy, authority, etc.)
    - Find new opportunities for mitigations

What Happens When Risk is Realized?

With Trust*:

- Team conveys to stakeholders
- Team is allowed to solve the issue
- Opportunity to have conversation about approach to problem solving
- Knowledge, wisdom, and earned trust is what can remove process

Without Trust:

- (Common) Knee jerk reactions
  - More oversight
  - More bureaucracy
- Process is a substitute for knowledge
  - Knowledge of the team OR stakeholders
- Pulls agency from capable people
- Generally slows and adds cost

When is it ok to allow Risk to be realized?

*Assumed to have more experience and good mentorship on team

What Happens When Risk is Realized?

- When can we set failure as expectation?
- What a program looks like that embraces failure so that you can take risk intelligently?
  - Not necessarily “constrained”, but purposeful risk taking
  - Design program around a team progressively building capability (vs. one-shot must-work programs)
    - Stakeholders accepting of types of failure
  - Scale amount of impact of a failure
    - Can fail under certain circumstances… “better to try than not to do” (a la Mike Swartwout)
    - AFRL
      - Doesn’t allow bus as experiment any more (Safety concerns)
      - Does allow experiment to not work or be different than expected (i.e. its an experiment)

Approaches

Constraints

Schedule constrained

Schedule drives scope

- A faster cadence to orbit is more desirable to achieve MVP than an exquisite system in 5+ years
- Attempting to move faster informs program scope, which may change to achieve schedule

Talk with stakeholders on what they care about & how to approach problems

Budget Constrained

Budget drives scope

- Additional capabilities, reliability/robustness, performance may be traded to reduce costs
- Attempting to stay within budget informs program scope, which may change to achieve budget

Higher Risk Acceptance

Variable confidence in capability

- Generally project team accepts risks towards achieving full success while stakeholders may accept risks towards achieving minimum success.
- May accept risk towards robustness, systems are generally single string with minimal designed radiation tolerance at a parts level.

Risk Mitigation Conversation

Baseline "Capability"

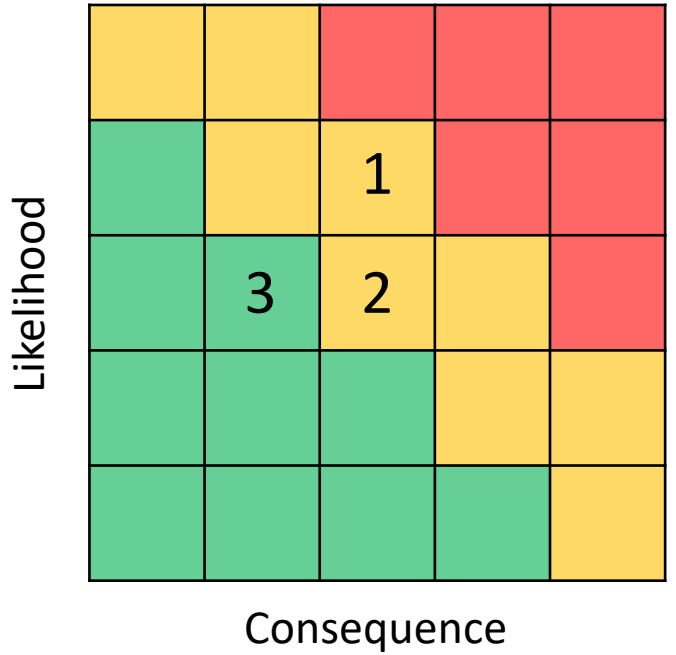
- Nominal scope w/in objective set
- Associated "nominal" practices associated with cost/schedule profile and capability

Line moves based on priorities and resources

Designed Capability/Scope (e.g. – PLD)	Risk Mitigation (e.g. Radiation)	Verification (e.g. SV Functional)	Operations (e.g. ease-for-operator, example only)
❑ MVP	❑ SV software radiation considerations	❑ Cmd Ex	❑ Editable command schedule
❑ File transfer	❑ Rad tol. memory	❑ DitL	❑ Commanding automation
❑ Tlm stream	❑ All rad tol. parts	❑ Long-range	❑ Commanding visualization
❑ Cmd over ntwk	❑ Redundancy	❑ LV DNH	❑ …
❑ Tlm receive	❑ …	❑ FFT	
❑ Store and fwd		❑ Fault Test	
❑ …		❑ …	
❑ ASI		❑ ADCS V&V	
		❑ EMC/EMI	

A Simple Approach to Risks

#	Description	Mitigation
1	If <event>, then outcome/consequence	Method(s) to address
2	If <event>, then outcome/consequence	Method(s) to address
3	If <event>, then outcome/consequence	Method(s) to address



Likelihood assessed by project team, follows more informal levels of:

- Probably wont happen
- Possible
- Will occur

Consequence Scale

1 – Full success achievable	2 – Affects Some Full Success	3 – Minimum Success Achievable	4 - Affects Minimum Success / Major Delay	5 – DOA / no flight

Key Threads to Consider

- Unfortunately I don't fully have answers…
- Can we define a "standard practice" for when a team can/should take risks?
  - When is a team mature enough to do this effectively?
  - What is the consequence when something goes wrong?
    - "Did we use our money + time correctly?"
    - What is the stakeholder environment and how do you earn their trust up & down?
  - Can we identify situations when we SHOULD take risks?
    - What opportunities would we get?
- Can we turn risk taking activities into research?
  - How to take risk? (Trust v. Technical v. etc.)
  - How do different people / industries view risk? (Investors, Medical, etc.)